

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A method for operating a code cache in a dynamic instruction translator comprising the steps of:
 - storing a plurality of translations in a cold partition in a cache memory;
 - determining whether a translation that has been stored in the cold partition is hot; and
 - moving the translation to a hot partition in the cache memory when a translation has been determined to be hot.
2. (Original) A method as defined in claim 1, wherein the step of determining whether a translation is hot comprises:
 - maintaining a different associated counter for each of a plurality of translations in the cold partition of the cache memory;
 - incrementing or decrementing the count in the associated counter each time its associated translation is executed; and
 - concluding the determination that a translation is hot if the count in the associated counter reaches a first threshold value.
3. (Original) A method as defined in claim 1, wherein said hot partition is contiguous and disjoint from said cold partition in said cache memory.
4. (Original) A method as defined in claim 2, wherein said maintaining an associated counter step comprises maintaining counters in a data structure external to said cache memory.

5. (Original) A method as defined in claim 4, further comprising the step of at least temporarily delinking blocks of translations stored in said cold partition so that control exits the cache memory in order to perform the incrementing or decrementing step.

6. (Original) A method as defined in claim 2, wherein said maintaining within said cache memory an associated counter step comprises maintaining one of said associated counters for each entry point into a plurality of the translations in said cold partition of the cache memory.

7. (Original) A method as defined in claim 2, wherein said maintaining an associated counter step comprises logically embedding update code on an arc between two translations.

al 8. (Original) A method as defined in claim 2, wherein said maintaining an associated counter step comprises maintaining one of said associated counters for each machine cache line in an associated microprocessor.

9. (Original) A method as defined in claim 2, wherein said translation moving step comprises sampling a plurality of said associated counters on an intermittent basis to determine if the count therein has reached said threshold value.

10. (Original) A method as defined in claim 1, further comprising the steps of:
determining if a number of hot translations in said hot partition of said cache memory exceeds a second threshold value; and

if said number of said hot translations exceeds said second threshold value, then expanding the size of said hot partition in said cache memory by adding thereto an expansion area contiguous to said hot partition.

11. (Original) A method as defined in claim 10, further comprising the step of

removing all cold translations from said expansion area and storing said removed translations in said cold partition.

12. (Original) A method as defined in claim 2, wherein the maintaining an associated counter step comprises maintaining an associated counter for all translations in the cold partition of the cache memory.

13. (Original) A system for a code cache in a dynamic instruction translator comprising:
a cache memory;
a cold partition and a hot partition in said cache memory;
logic for determining whether a translation that has been stored in the cold partition is hot; and
logic for moving the translation to a hot partition in the cache memory when a translation has been determined to be hot.

14. (Original) A system as defined in claim 13, wherein the logic for determining whether a translation is hot comprises:
logic for associating a different counter for each of a plurality of translations stored in the cold partition of the cache memory;
logic for incrementing or decrementing the count in the associated counter each time its associated translation is executed; and
logic determining if the count in the associated counter reaches a first threshold value.

15. (Original) A system as defined in claim 13, wherein said hot partition is contiguous and disjoint from said cold partition in said cache memory.

16. (Original) A system as defined in claim 14, wherein said counters are maintained in a data structure external to said cache memory.

17. (Original) A system as defined in claim 16, wherein said incrementing or decrementing logic further comprises logic for at least temporarily delinking blocks of translations stored in said cold partition so that control exits the cache memory in order to perform the incrementing or decrementing of the count.

18. (Original) A system as defined in claim 14, wherein said logic for associating counters comprises logic for maintaining one of said associated counters for each entry point into a plurality of the translations in said cold partition of the cache memory.

19. (Original) A system as defined in claim 14, wherein said logic for moving the translation comprises logic for sampling a plurality of said associated counters on an intermittent basis to determine if the count therein has reached said threshold value.

20. (Original) A system as defined in claim 13, further comprising:
logic for determining if a number of hot translations in said hot partition of said cache memory exceeds a second threshold value; and
if said number of said hot translations exceeds said second threshold value, logic for expanding the size of said hot partition in said cache memory by adding thereto an expansion area contiguous to said hot partition.

21. (Original) A system as defined in claim 20, further comprising:
logic for removing all cold translations from said expansion area and storing said removed translations in said cold partition.

22. (Original) A system as defined in claim 14, wherein the logic for associating a counter step comprises logic for maintaining an associated counter for all translations in the cold partition of the cache memory.

23. (Currently Amended) A program product, comprising a computer usable medium having computer readable program code embodied therein that, when executed, directs ~~for directing~~ a computer to manage a cache memory by:

storing a plurality of translations in a cold partition in a cache memory;

determining whether a translation that has been stored in the cold partition is hot; and

moving the translation to a hot partition in the cache memory when a translation has been determined to be hot.
